

①9 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

①1 N° de publication :

2 774 191

(à n'utiliser que pour les
commandes de reproduction)

②1 N° d'enregistrement national :

98 01725

⑤1 Int Cl⁶ : G 06 F 15/16

⑫

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 28.01.98.

③0 Priorité :

④3 Date de mise à la disposition du public de la
demande : 30.07.99 Bulletin 99/30.

⑤6 Liste des documents cités dans le rapport de
recherche préliminaire : *Ce dernier n'a pas été
établi à la date de publication de la demande.*

⑥0 Références à d'autres documents nationaux
apparentés :

⑦1 Demandeur(s) : INSTITUT EURECOM Groupement
d'interet economique — FR.

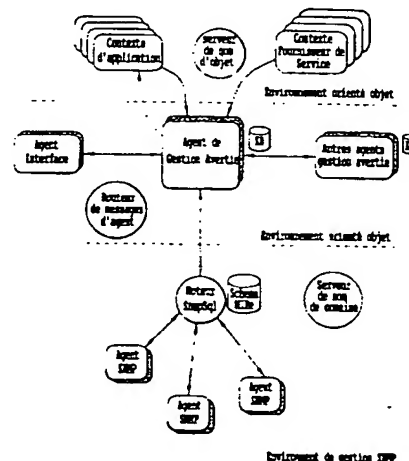
⑦2 Inventeur(s) : DE OLIVEIRA RAOUL TEXEIRA et
LABETOULLE JACQUES.

⑦3 Titulaire(s) :

⑦4 Mandataire(s) : NONNENMACHER BERNARD.

⑤4 PROCÉDE D'ADMINISTRATION DE RESEAUX A L'AIDE D'AGENTS INTELLIGENTS.

⑤7 L'invention concerne un procédé d'administration de réseau faisant intervenir une pluralité d'utilisateurs du réseau et une pluralité de ressources et/ ou de services disponibles dans le réseau et destinées à être utilisées par lesdits utilisateurs, caractérisé en ce qu'il comporte des étapes consistant à informer le système de gestion de réseau des besoins des utilisateurs, et à informer les utilisateurs des ressources et/ ou services disponibles dans le réseau, de sorte que ledit système de gestion de réseau puisse de façon autonome exécuter des opérations de gestion aptes à optimiser le fonctionnement du réseau en fonction des besoins des utilisateurs et des ressources et/ ou services disponibles dans le réseau.



FR 2 774 191 - A1



La présente invention concerne les procédés et dispositifs de gestion de réseaux d'ordinateurs, et plus particulièrement la gestion de la qualité de service dans un réseau.

Plus précisément, l'invention concerne un procédé permettant d'introduire et de prendre en compte les besoins des utilisateurs de réseaux, aussi bien utilisateurs finaux que fournisseurs de services, dans une structure de gestion de réseau, afin de permettre à des applications de gestion de réseau de décider en direct des opérations appropriées de gestion de réseau, sans intervention humaine.

Etat de la Technique:

L'interconnexion mondiale des ordinateurs est aujourd'hui en passe de devenir une réalité. Les réseaux font et feront de plus en plus partie de l'environnement quotidien, et permettront de gérer des applications pour une variété croissante de tâches, allant par exemple jusqu'à la gestion télécommandée de l'équipement ménager et électrique des maisons. De tels réseaux, en tant que systèmes distribués, nécessitent surveillance et administration. Or les réseaux privés, comme par exemple les réseaux locaux ou ceux des petites entreprises, n'ont pas les mêmes besoins en terme d'administration que les réseaux à l'échelle mondiale. Pourtant, si la plupart des périphériques partageables en réseau sont administrables (via SNMP, pour « Simple Network Management Protocol » en terminologie anglosaxonne), seules quelques plateformes d'administration existent, spécifiquement conçues pour les grands réseaux.

La gestion de réseaux et la gestion de systèmes (distribués ou non) ont été pendant longtemps des activités différentes et séparées, même si elles présentaient des interactions. Récemment, les deux activités ont connu une certaine intégration, de sorte qu'il est maintenant possible à partir d'une plateforme de gestion unique de gérer de façon homogène des réseaux et des systèmes. Des exemples de ce type d'intégration sont fournis par l'intégration de NETVIEW (solution de gestion de réseau commercialisée par IBM) avec TME10 (gestion de systèmes distribués, de TIVOLI), ou par OPEN VIEW (solution de gestion de réseau de Hewlett Packard) avec TNG Unicenter (gestion de systèmes distribués de Computer Associates). D'autres plateformes similaires existent sur le marché, mais aucune n'intègre une forme de sensibilité ou de « conscience » des besoins des utilisateurs finaux, c'est à dire des profils d'utilisateurs exprimant la façon dont l'utilisateur prévoit d'utiliser un système distribué.

Inconvénients de l'état de la Technique:

L'absence de communication entre les applications d'administration et les systèmes d'administration est la clé d'un problème majeur et encore sans solution à ce jour en matière d'administration de réseaux.

5 Ainsi, dans les plateformes d'administration de réseau disponibles aujourd'hui, toutes les fonctions de gestion de réseau ne sont pas couvertes, car les plateformes existantes se focalisent sur la simple surveillance du réseau et sur le rapport d'évènements.

En outre, aucune application d'administration de réseau connue n'est capable de gérer efficacement un réseau hétérogène. De plus, les produits existants, bien que de
10 fonctionnalité limitée, sont chers et consommateurs de ressources et ne sont pas adaptés à de petits réseaux.

Enfin, dans les systèmes et procédés d'administration de réseaux de l'état de la technique, il n'est pas tenu compte des services qui seront utilisés par les applications installées sur le réseau, ni surtout d'un niveau de qualité de service attendu par un
15 utilisateur ou une application. De ce fait, les applications, qui pourtant dépendent de plus en plus du réseau, ont difficilement accès à des informations le concernant.

Une application a rarement la possibilité de déterminer à l'avance si les ressources qu'elle requiert sont disponibles. Par exemple, une application réseau qui repose sur un système de fichiers distribué n'a aucun moyen de savoir si la qualité de
20 services requise n'est pas en train de se dégrader. De ce fait, il arrive que la moindre dégradation des ressources du réseau peut dramatiquement affecter la stabilité de l'application, voire l'empêcher totalement de fonctionner.

Arrière plan de l'invention:

En raison des différents inconvénients mentionnés ci-dessus, une nouvelle
25 approche de gestion qui est déjà souhaitable aujourd'hui dans les environnements complexes de réseaux, deviendra vitale dans le futur. En fait, il est probable qu'à mesure où l'usage de l'ordinateur de réseau (NC: Network Computer en terminologie anglosaxonne) tend à se développer, le réseau lui-même va revêtir une importance de plus en plus grande pour les utilisateurs finaux, et réciproquement.

30 Ainsi, il va devenir impossible de concevoir une infrastructure de gestion de réseau qui ne prenne pas en compte les besoins des utilisateurs finaux, puisque ces derniers

représentent la plus importante source d'information pour conserver le fonctionnement optimal d'un environnement de réseau.

Or ce fonctionnement dépend à la fois des besoins des utilisateurs en termes de paramètres de qualité de service, que des caractéristiques de l'offre de service disponible dans le réseau pour les utilisateurs à chaque instant. Et il est clair que les problèmes d'administration liés à l'apparition d'applications mobiles ne peuvent plus être traités efficacement par les anciennes approches de gestion statique et centralisée des réseaux

Buts de l'invention:

Le but principal de l'invention est par conséquent de proposer un procédé et des mécanismes de gestion de réseau permettant de résoudre l'ensemble des inconvénients des systèmes d'administration de réseau connus dans l'état de la technique. Plus particulièrement, un des buts de l'invention est de proposer un procédé d'administration de réseau simple, permettant une meilleure intégration et collaboration entre les applications, et des plateformes d'administration extensibles et peu coûteuses. Un autre but de l'invention est de proposer un procédé d'administration permettant de remplir automatiquement des tâches d'administration aptes à prendre en compte les besoins des utilisateurs du réseau.

Un autre but de l'invention est de proposer une plateforme d'administration de réseau qui soit à la fois flexible et extensible pour pouvoir prendre en compte l'ajout de ressources ou de services, et économique.

D'autres buts et avantages de l'invention vont apparaître à la lecture de la description suivante faite à titre d'exemple non limitatif, et aux dessins ci-annexés, dans lesquels:

- La figure 1 représente un environnement réseau du type de ceux où le procédé de gestion selon l'invention peut être mis en oeuvre,

- La figure 2 représente sous forme schématique une architecture de gestion y compris les principaux agents permettant de mettre en oeuvre l'invention,

- La figure 3 représente sous forme schématique les éléments d'un contexte d'application utilisé pour mettre en oeuvre le procédé selon l'invention,

introduire tableau page 92

- La figure 4 représente sous forme schématique les éléments d'une contexte de fournisseur de service utilisé pour mettre en oeuvre le procédé selon l'invention,

- La figure 5 représente un contexte d'application et un contexte de fournisseur de service, pour deux types différents de services: l'impression et le serveur de fichiers;

- La figure 6 représente sous forme schématique la structure d'un agent intelligent pouvant être utilisé pour mettre en oeuvre le procédé selon l'invention;

5 - La figure 7 représente l'intégration d'agents intelligents dans un exemple d'environnement de réseau;

Principe de la solution:

L'invention décrite plus en détail ci-dessous est la première tentative de gestion intégrant non seulement la gestion de réseaux et de systèmes distribués, mais va plus loin
10 en intégrant également des informations concernant la façon dont les utilisateurs prévoient d'utiliser des services disponibles dans un environnement de réseau d'entreprise, par exemple comme celui représenté en figure 1. Selon l'invention, il est proposé une méthode et des outils pour introduire dans l'infrastructure de gestion de réseau, une connaissance ou une « conscience » des besoins des utilisateurs finaux et des ressources des fournisseurs de
15 services, afin de permettre de décider à la volée des opérations appropriées de gestion de réseau, telles que la surveillance, la commande, le diagnostic, ou la reconfiguration du réseau, et ce sans aucune intervention humaine.

Dans la suite, on définira par mesure de simplicité la gestion de réseau avec connaissance des besoins des utilisateurs par la terminologie de « gestion avertie », qui
20 serait mieux représentée en terminologie anglosaxonne par l'expression « aware management ».

L'invention concerne un procédé d'administration de réseau faisant intervenir une pluralité d'utilisateurs du réseau et une pluralité de ressources et/ou de services disponibles dans le réseau et destinées à être utilisées par lesdits utilisateurs, caractérisé en
25 ce qu'il comporte des étapes consistant à informer le système de gestion de réseau des besoins des utilisateurs, et à informer les utilisateurs des ressources et/ou services disponibles dans le réseau, de sorte que ledit système de gestion de réseau puisse de façon autonome exécuter des opérations de gestion aptes à optimiser le fonctionnement du réseau en fonction des besoins des utilisateurs et des ressources
30 et/ou services disponibles dans le réseau.

Selon d'autres caractéristiques du procédé de gestion selon l'invention:

- il comporte des étapes consistant à:

- rassembler les besoins des utilisateurs à partir des applications utilisatrices de ressources, et les capacités de prestations des ressources et/ou services à partir des applications prestataires de ressources et/ou de services;
- transmettre lesdits besoins et capacités des applications à un
5 environnement de gestion de réseau;
- au niveau de l'environnement de gestion de réseau, traiter les informations reçues et décider des opérations de gestion à effectuer pour que l'environnement de réseau fonctionne comme requis par les applications utilisatrices;
- au cas où l'environnement de réseau ne peut pas fonctionner comme
10 demandé, en informer les applications utilisatrices et/ou les applications prestataires.
- pour rassembler les besoins des utilisateurs à partir des applications utilisatrices de ressources, on utilise une structure d'information dynamique QdS représentative des besoins de Qualité de Service (QdS), comprenant un en-tête de contexte d'application et un bloc d'information représentatif des spécifications de qualité
15 de service pour chaque service parmi une pluralité de services.
- ledit bloc d'information représentative des besoins de qualité de service comporte une séquence de dimensions de QdS, chaque dimension de QdS contenant une séquence de Domaines, et chaque Domaine contenant une séquence d'attributs de QdS.
- pour rassembler les capacités de prestations des ressources et/ou services à
20 partir des applications prestataires de ressources et/ou de services, on utilise une structure d'information dynamique représentative des offres de qualité de service à l'attention des utilisateurs, comprenant un en-tête de contexte de fournisseur de service, un bloc d'information représentatif des spécifications de qualité de service offertes par chaque service disponible, et un un bloc d'information représentatif des spécifications de
25 qualité de service pour chaque service parmi une pluralité de services.
- pour traiter les informations reçues et décider des opérations de gestion à effectuer, on élabore des tâches de gestion abstraites à l'aide d'agents autonomes qui prennent en compte les besoins de qualité de servie QdS des utilisateurs, et les offres de service dans l'environnement du réseau.
- de préférence, lesdits agents sont des agents intelligents, comportant :
30
 - une couche avertie permettant d'acquérir les besoins des utilisateurs du réseau et les offres de services;

- une couche délibérative chargée de créer des tâches de gestion;
- une couche opérationnelle exécutant les tâches de gestion créées par la couche délibérative.

Selon un aspect de l'invention, la saisie des besoins des utilisateurs se fait par l'intermédiaire d'une interface homme-machine comprenant un moyen d'entrée d'information et un moyen d'affichage.

Description détaillée des figures:

On se réfère à la figure 1. On a représenté dans cette figure un exemple de structure de réseau permettant de mettre en oeuvre l'invention. L'environnement 1 représenté est composé de plusieurs domaines utilisateurs, tel que celui du côté gauche de la figure, un domaine de serveurs 2 où sont situés les serveurs d'entreprise, et un réseau principal 3 permettant de fédérer toutes les communications soit pour accéder à des ressources du réseau dans l'entreprise ou à l'extérieur.

Les utilisateurs finaux ou les applications expriment leurs besoins par l'intermédiaire d'une structure d'information qui sera appelée dans la suite un contexte d'application ou Desiderata de Qualité de Service (QdS). Cette structure d'information détaille les services et les attributs de QdS associés qui sont considérés appropriés pour assurer un comportement correct de la part d'une application. Un Desiderata de QdS est l'élément qui permet à une plateforme de gestion d'atteindre une connaissance concernant les services en cours d'utilisation, leur utilisateur, et la façon dont ces services sont utilisés. De ce fait, il est également possible, grâce à l'invention, d'en déduire le degré de satisfaction des utilisateurs finaux de l'environnement de réseau. Il doit être rappelé que ce qui a été jusqu'à présent évalué dans l'état de la technique, n'est que le degré de satisfaction du fournisseur de services, au lieu du degré de satisfaction des utilisateurs finaux.

La définition d'une structure capable de porter une combinaison de requêtes pour de multiples services ayant chacun ses caractéristiques propres, implique la conception d'une structure d'information dynamique pour les Desiderata de QdS.

On se réfère à la figure 2, qui représente de façon schématique l'architecture d'un ensemble de gestion de réseau intégrant les principaux éléments nécessaires pour la mise en oeuvre du procédé selon l'invention.

Cette architecture se focalise sur les agents AIs de QdS. Les AIs de QdS reçoivent, via une couche CORBA les besoins des applications (Application Contexts) et des fournisseurs de service (Service Provider Contexts). Ils dialoguent ensuite (toujours via CORBA) avec un serveur SQL qui joue le rôle d'interface entre l'AI et des agents
5 SNMPS classiques. Ce serveur traduit les requêtes de l'AI (exprimées en SQL) en requêtes de type SNMP à l'attention des agents SNMP, recueille les résultats et les communique (sous forme SQL) à l'AI. Pour de nombreuses raisons, un AI peut avoir à communiquer avec un autre AI. Dans ce cas, la communication se fait via des messages au format KQML encapsulant des messages spécifiés soit en KQML, soit en GDL. Un
10 agent Interface d'Agent a été conçu pour assister les managers humains pour communiquer avec les agents intelligents, à l'aide d'une interface graphique d'utilisateur (voir plus loin).

L'Agent Interface est un agent sous forme d'applet qui permet à un manager a priori humain d'interroger et d'interagir avec des Agents Intelligents. La communication
15 proprement dite se fait au moyen de messages par exemple au format KQML.

Le routeur de messages inter-agents, ou AMR (Agent Message Router) est une application spécialisée (un assistant de communication), qui à la réception d'un message provenant d'un agent, le redirige vers l'agent destinataire. Les messages reçus sont gérés dans une file d'attente conservée sur le système de fichiers.

20 Le serveur SQL joue le rôle d'un interprète entre AI et agents SNMP, le premier parlant SQL, les autres SNMP.

L'interface utilisateur est une interface sous forme d'applet (petite application) qui permet à un utilisateur de spécifier directement ses desiderata en matière de QdS. Cette interface n'a de vertu que démonstrative, puisqu'idéalement ces desiderata
25 seront essentiellement spécifiés directement par les applications.

La connaissance des besoins des utilisateurs est une notion qui est absente des systèmes connus de gestion de réseau. De ce fait, la gestion dite avertie est un effort pour donner un sens à cette expression dans le cadre de systèmes de gestion de réseau, de sorte que le résultat de la gestion du réseau devienne plus satisfaisante du point de vue d'un
30 utilisateur final.

On définit par conséquent la « gestion avertie » comme étant la capacité d'un système de gestion de décider de ses activités pour un environnement de réseau, sur la base des

besoins courants des utilisateurs. Cette gestion avertie prend en compte des questions comme: qui sont les utilisateurs finaux ? Quels sont leurs besoins courants de service ? comment sont-ils logiquement organisés ? Où sont-ils situés, etc.

Les besoins des utilisateurs et les offres de services devant être pris en compte, il devient nécessaire de trouver une façon d'exprimer les besoins des utilisateurs et les offres de service à l'aide de structures d'information respectives appropriées. Ces structures d'information sont décrites en relation avec les figures 3 et 4. De façon générale, chaque entité de QdS doit être identifiée par un nom unique, qui est composé de son propre nom plus une concaténation des noms de toutes les entités QdS qui la contiennent. Ainsi, le nom d'un Domaine de QdS doit être du type [Nom de Service][Nom de Dimension QdS][Nom de
10 Domaine QdS]. Les besoins de QdS et les entités de service possèdent également un en-tête d'entité contenant des informations génériques par rapport à eux-mêmes.

Toutes les spécifications de QdS doivent suivre les recommandations ci-dessus.

On se réfère à la figure 3 pour décrire la structure d'un besoin ou desiderata de QdS. Il contient les spécifications de service de chaque service que l'application requiert pour
15 fonctionner correctement. Une spécification de service contient elle-même une liste de conditions (contraintes) implicites sur des attributs (paramètres) de QdS associés au service. Ces attributs de QdS sont logiquement organisés de façon hiérarchique en Dimension de QdS et Domaine de QdS. Toutes les spécifications de QdS doivent suivre
20 ces règles. L'avantage d'une telle approche est qu'elle permet à l'Agent de travailler à un niveau élevé d'abstraction. Ainsi un Agent peut se voir amené à prendre une décision en se basant seulement sur la Dimension de QdS plutôt que sur chaque attribut de QdS de chaque Domaine de QdS de la Dimension de QdS

Les besoins de QdS sont à la base d'une gestion avertie, au sens de l'invention
25 Une telle entité comporte toute l'information concernant tous les services utilisés, comment et par qui ils sont utilisés, et avec quel niveau de priorité.

La structure d'un contexte d'application (ou besoin de QdS) comporte un *en-tête de contexte d'application*, où des informations générales concernant l'application d'utilisateur final sont stockées, et des besoins de services, dans lesquels sont stockées
30 les spécifications détaillées de chaque service. La table 1 ci-dessous donne le détail de ce premier niveau de contexte d'application:

Table 1:

| Nom | Signification |
|-----------------------------------|--|
| En-tête de contexte d'application | Description générale de l'application présente dans l'environnement de réseau |
| Besoins de Services | Bloc d'informations où sont inclus les spécifications de QoS pour chaque service |

La table 2 représente les attributs de l'en-tête de contexte d'application qui identifie l'application d'utilisateur final et l'utilisateur qui utilise l'application.

5 Table 2:

| Nom | Signification |
|------------------------------|---|
| Nom d'Application | Nom que l'éditeur de logiciel a donné à l'application |
| Type d'application | Caractérisation du rôle de l'application dans l'environnement de réseau (session d'utilisateur) |
| Identification d'Application | Distingue deux applications du même type lancées dans le même hôte. |
| Nom d'utilisateur | Nom de l'utilisateur qui est responsable de cette application ou est en train de l'utiliser |
| Identifiant d'utilisateur | Identifiant de l'entreprise de l'utilisateur |
| Nom de Host | Nom du noeud de réseau où l'application est située |
| Priorité | Priorité donnée par l'utilisateur à cette application |

Les besoins de service sont composés d'une séquence de contextes de service, représentés dans la table 3.

10 Table 3: Principaux blocs d'un contexte de service

| Nom d'attribut | Signification |
|--------------------------------|--|
| En-tête de contexte de service | attributs génériques concernant un service sélectionné |

| | |
|-------------------------------|---|
| Séquence de dimensions de QdS | les dimensions de QdS constituent le premier niveau de détail ayant trait à des demandes d'utilisateurs pour un service donné |
| Séquence de Ressources | Ressources de service requises par l'application de ce service |

Table 4: attributs définis dans l'en-tête de contexte de service

5

| Nom d'attribut | Signification |
|--------------------------------|---|
| En-tête de contexte de service | attributs génériques concernant un service sélectionné |
| Séquence de dimensions de QdS | les dimensions de QdS constituent le premier niveau de détail ayant trait à des demandes d'utilisateurs pour un service donné |
| Séquence de Ressources | Ressources de service requises par l'application de ce service |

Table 5: Attributs définis dans le contexte de service

| Nom | Signification |
|------------------------|--|
| Service | nom généralement accepté pour le service |
| Fournisseur de Service | nom du fournisseur de service, quelquefois identique au nom du système qui l'héberge |
| Priorité | Priorité du service pour l'application qui l'utilise |
| Media | décrit le type de média dont le service a besoin, tel que données, vidéo, audio, etc. |
| Engagement | décrit la classe d'engagement que l'application utilisant ce service attend à partir de la plateforme de gestion, tel que meilleurs efforts, statistique, déterministe |

10

On se réfère à la figure 4 pour décrire la structure d'information représentative du
 15 contexte de fournisseurs de service. Les Dimensions de QdS et Domaines de QdS

spécifiés dans le Contexte d'application correspondent évidemment à ceux définis par les fournisseurs de service. Ces derniers expriment leur capacités dans des Contextes de fournisseurs de service, qui contiennent des attributs de QdS (hiérarchiquement organisés en Dimension de QdS et Domaines de QdS) que le fournisseur de service est capable d'offrir. Les fournisseurs sont responsables de l'instrumentation du service, i.e doivent non seulement spécifier les attributs de QdS les plus appropriées pour le service, mais aussi les outils de mesure permettant de les surveiller.

La figure 5 récapitule sous forme schématique les structures d'information représentatives d'un contexte d'application et du contexte de fournisseur de service, pour deux exemples de services souvent utilisés en pratique: un service de serveur de fichiers (noté DFS), et un service d'impression (noté DPRINTING).

Le procédé de gestion selon l'invention et les entités et structures d'information utilisés peuvent être de préférence mis en oeuvre à l'aide d'agents autonomes intelligents, notés AI. L'AI est une composante centrale de la stratégie de gestion. Ses propriétés sont les suivantes:

Réactif, dès l'apparition d'un problème l'agent doit définir les mesures appropriées à prendre et éventuellement employer les outils à sa dispositions pour agir sur la source du problème.

Proactif, il doit anticiper l'apparition des problèmes et chercher à améliorer les performances du système avant que les valeurs critiques des seuils ne soient atteints.

Autonome, il doit posséder une certaine autonomie de décision quant aux réponses à apporter en cas de pannes, et ne notifier le manager que lorsque le problème dépasse ses compétences. En revanche, il doit toujours notifier les applications des moindres dégradations voire interruptions d'un service dont il garantit la QdS.

La figure 6 donne une représentation schématique d'un mode de réalisation d'un agent intelligent pouvant être utilisé pour mettre en oeuvre le procédé de gestion selon l'invention. On peut distinguer 3 couches dans l'architecture d'un Agent Intelligent (AI).

La première couche, que l'on appellera la «partie sensible» (awareness layer en terminologie anglo-saxonne), se compose essentiellement du Gestionnaire de contextes (Context Manager). Ce Gestionnaire offre plusieurs interfaces, accessibles par tous les acteurs définis plus haut, à savoir les applications utilisateurs, les fournisseurs de service

et les applications d'administration. Grâce à ces interfaces, le Gestionnaire de contextes peut gérer des informations orienté services de haut niveau.

-La «partie délibérative» (deliberative layer) forme la seconde couche, où l'on retrouve principalement le Gestionnaire de buts d'administration (Goal Manager).

5 -La «partie exécutive» (operational layer) est la troisième et dernière couche, et est formée par les procédures d'administration. Chaque but d'administration de la couche supérieure est associée à une intention d'administration qui représente les procédures d'administration décidées par l'AI.

10 On va maintenant décrire de façon plus détaillée chacune des trois parties des agents intelligents utilisés dans le cadre de l'invention:

1) Partie avertie:

L'Agent Intelligent (AI) étant typiquement destiné à être placé dans un environnement distribué, on doit par conséquent le doter de mécanismes lui permettant de détecter les caractéristiques de son environnement, et en particulier, dans le cas présent, les
15 desiderata ou besoins des applications.

Comme il est aujourd'hui attendu que les systèmes distribués du futur seront bases sur un paradigme de communication via un «langage neutre», tel que CORBA, l'AI doit lui aussi proposer des interfaces, spécifiés en CORBA IDL, aux applications, fournisseurs de services et systèmes d'administration.

20 Ces interfaces fournissent des méthodes pour la négociation, publication et mise à jour des contextes. Les manipulateurs de contextes (Service, Dimension de QdS et Domaine de QdS) analysent les contextes reçus et les stockent dans des Bases de contextes. Ils passent ensuite une référence des contextes fraîchement reçus au Gestionnaire de buts d'administration.

25 Le Gestionnaire de contextes - « Context Manager »

Le gestionnaire de contexte est chargé de collecter les desiderata des applications et fournisseurs, sous la forme de contextes d'applications et de contextes de fournisseurs de service, de les interpréter et de les passer à la couche inférieure.

2) La Partie délibérative:

30 - Gestionnaire de buts d'administration « Goal Manager »

A la réception d'un contexte d'application (via la couche supérieure), ou d'un but d'administration d'un tiers agent (via le module de communication), le Gestionnaire de

but d'administration doit décider si un but d'administration doit localement être créé ou non, en se basant entre autres choses sur un ensemble de politiques définies par les administrateurs de réseau. S'il décide de le créer, il le signale au module «Mémoire», qui le conservera dans sa Base de catégories mentales. Ainsi l'AI pourra vérifier au besoin si un but d'administration n'est pas déjà planifié.

Une autre décision est alors à prendre par l'AI: faut-il traiter localement le but d'administration, le déléguer à un autre AI, ou les deux à la fois? En fait, la décision de déléguer un but sera prise si l'agent ne peut accéder à certaines ressources du système distribué car ne se trouvant pas dans son domaine, ou si l'AI estime ne pas être suffisamment efficace pour traiter le but demandé, car par exemple trop éloigné des ressources à gérer. Une fois de plus, cette décision est basée entre autres sur des politiques définies par les administrateurs de réseaux.

Module de communication - »Conversation Dispatcher » Le gestionnaire de buts d'administration implémente une interface dédiée aux échanges de catégories mentales. Cette interface doit être accessible à tous les agents ~puisque'elle autorise les autres agents à déléguer un but d'administration, ou encore à obtenir des informations sur un but d'administration et les outils de mesure de service associés.

Décision de création de but d'administration: Un but d'administration comporte un ensemble de conditions («desiderata») portant sur des attributs de QdS. Cette information est directement associée aux conditions définies par un contexte d'application. Ce dernier peut donner lieu à un ou plusieurs buts d'administration, puisqu'il peut y avoir plusieurs ensembles d'attributs de QdS (i.e des Domaines de QdS), et qu'un but d'administration dépend au moins d'un domaine de QdS (il sera par la suite traduit en procédures d'administration).

En se basant sur le contexte d'application (nom du service, les clients, les fournisseurs, ainsi que les attributs, domaines et dimensions de ~QdS) et sur les politiques du manager, le Cerveau doit alors décider si oui ou non la création d'un nouveau but d'administration s'avère nécessaire.

Pour une implémentation possible, on peut décider d'implémenter la règle suivante: *l'AI décide de créer* un nouveau but:

- a) s'il n'y a aucun but d'administration déjà présent dans la Base de catégorie mentale comportant le triplet (nom du service, dimension de QdS, domaine de QdS)

ou bien:

b) si un tel triplet existe, mais que l'accomplissement du but associé n'implique pas l'accomplissement du nouveau but.

On peut voir le triplet ci-dessus comme un index global de but d'administration. L'AI peut utiliser cet index lors de la création d'un nouveau but pour rechercher dans la Base de catégorie mentale si des buts similaires ne seraient pas déjà en cours. C'est l'information minimale (extraite du contexte d'application) qui permet de distinguer deux buts similaires. Bien sûr, cet index dépend de l'interprétation de l'information contenue dans le but...

L'implication b) est un des éléments qui font l'intelligence de l'AI. Il dit que si un but doit être accompli, tout autre but impliqué par lui n'a pas besoin d'être accompli. Par exemple, si une application requiert un temps de réponse de 10ms pour le service NFS (serveur de fichiers), et qu'il existe déjà un but qui accomplit 5ms, le premier but est impliqué par le but déjà existant et n'a par conséquent pas besoin d'être créé. Cette règle est clairement spécifique à un but, i.e la définition de cette implication peut être différente (et différemment implémentée) pour chaque combinaison de l'index...

Décision de délégation de but d'administration Le modèle de décision vérifie si le fournisseur de service requis est bien situé dans le domaine géré par l'Agent. L'Agent Intelligent est conscient de la présence de fournisseurs de services puisqu'il reçoit des contextes de service de la part de chaque fournisseur de service. Par conséquent la décision à prendre est implémentée en recherchant dans la base des contextes de service le nom du fournisseur de service:

- si le fournisseur de service se situe dans le domaine géré par l'agent, le but sera traité localement, puisque l'agent est capable de récupérer les outils qui instrumenteront le service ;

- si le fournisseur de service appartient à un domaine de gestion sur lequel l'agent n'a aucun droit, le but doit être délégué.

Par exemple, si une application a besoin d'accéder au serveur Web d'un autre domaine de gestion avec un temps de réponse donné, au moins deux Agents (l'Agent gérant le domaine où se trouve l'application, et celui gérant le domaine où se situe le serveur Web) devront coopérer (dans le sens où ils devront travailler de concert) pour atteindre l'objectif demandé.

Implémentation Le « Centre d'Operations » est en fait initialement assez « stupide », mais est prêt à recevoir un moteur d'inférence ou tout autre système de manipulation de connaissances.

C'est au Centre de Décisions de gérer les priorités fixées par les managers, applications et fournisseurs de services, et les conflits qui peuvent en résulter. Par exemple, un fournisseur d'accès Internet souhaitera la plus haute qualité possible pour le service WEB, tandis qu'un étudiant imprimant son rapport de stage privilégiera pour un temps le service d'impression **DPRINTING**, pour rebasculer sur le service WEB une fois le rapport imprimé.

C'est encore au Centre de Décisions de décider de fusionner ou non des buts de gestions, ou encore de tenter de synthétiser ses croyances pour en dégager les croyances. Dans une implémentation possible, le Cerveau décide systématiquement de fusionner les buts. En ce qui concerne le second point, le Cerveau est aujourd'hui capable de synthétiser les périodes de non disponibilité d'un service donné, capacité qui peut facilement être étendue à des croyances plus générales.

Mémoire (Memory) La mémoire de l'AI est un module qui permet d'interroger et de manipuler la base de connaissance interne (KB) de l'AI. C'est elle qui conserve et maintient les catégories mentales de l'agent.

Implémentation La Mémoire est aujourd'hui implémentée sous la forme d'un tableau associatif complexe, qui permet d'associer à une clé soit une expression, soit un autre sous-tableau associatif complexe.

Module de communication (« Conversation Dispatcher ») Le module de conversation de l'AI est un module qui permet de communiquer avec d'autres agents.

25 3) La Partie exécutive

La partie exécutive se fonde essentiellement sur le module Effecteur (« Engine »). Il peut effectivement être comparé aux bras de l'AI, car il dispose de tous les éléments nécessaires pour mesurer la QdS, et au besoin de l'améliorer.

Procédures d'administration: Il y a trois types de procédures d'administration

- 30 - Surveillance et mesure
- Analyse et diagnostic
- Réparation et configuration

Implémentation On associe aux procédures d'administration des classes jouant le rôle d'outils de mesure de QoS (« *QoS Meter* »), d'outils de diagnostic - (« *QoS Diagnosis* »), et d'outil de configuration - (« *QoS Configure* »)

- 5 On se réfère à la figure 7 pour une illustration de l'utilisation d'agents intelligents en liaison avec un environnement de gestion de réseau donné.

Conclusion

- 10 L'invention propose pour la gestion des QoS une solution basée sur des Agents Intelligents (AIs), jouant le rôle d'assistants vis-à-vis des applications, et permettant ainsi le maintien d'une QoS la plus proche possible de la demande. Pour pouvoir gérer cette QoS individuellement par application, les AIs ont besoin de connaître leurs desiderata. Les structures d'information choisies à cette fin sont du type contexte, organisées suivant les divers services requis.

- 15 L'invention telle que décrite plus haut répond à ses objectifs, et permet entre autres de faire en sorte que les réseaux puissent "être au courant" des besoins et des comportements des applications, ce qui peut être considéré comme une forme d'« intelligence » à fort potentiel, non encore exploré. Cela permet d'informer les applications des dégradations de la qualité de service, et d'y faire face, par exemple par la
- 20 génération d'événements ou alarmes, non seulement en cas de pannes graves mais aussi dès que des dégradations de la qualité de service se font sentir de façon individuelle, ce qui est totalement impossible aujourd'hui.

REVENDEICATIONS

1. Procédé d'administration de réseau faisant intervenir une pluralité d'utilisateurs du réseau et une pluralité de ressources et/ou de services disponibles dans le réseau et destinées à être utilisées par lesdits utilisateurs, caractérisé en ce qu'il comporte des étapes consistant à informer le système de gestion de réseau des besoins des utilisateurs, et à informer les utilisateurs des ressources et/ou services disponibles dans le réseau, de sorte que ledit système de gestion de réseau puisse de façon autonome exécuter des opérations de gestion aptes à optimiser le fonctionnement du réseau en fonction des besoins des utilisateurs et des ressources et/ou services disponibles dans le réseau.

2. Procédé selon la revendication 1, caractérisé en ce qu'il comporte des étapes consistant à:

- rassembler les besoins des utilisateurs à partir des applications utilisatrices de ressources, et les capacités de prestations des ressources et/ou services à partir des applications prestataires de ressources et/ou de services;
- transmettre lesdits besoins et capacités des applications à un environnement de gestion de réseau;
- au niveau de l'environnement de gestion de réseau, traiter les informations reçues et décider des opérations de gestion à effectuer pour que l'environnement de réseau fonctionne comme requis par les applications utilisatrices;
- au cas où l'environnement de réseau ne peut pas fonctionner comme demandé, en informer les applications utilisatrices et/ou les applications prestataires.

3. Procédé selon la revendication 2, caractérisé en ce que pour rassembler les besoins des utilisateurs à partir des applications utilisatrices de ressources, on utilise une structure d'information dynamique QdS représentative des besoins de Qualité de Service (QdS), comprenant un en-tête de contexte d'application et un bloc d'information représentatif des spécifications de qualité de service pour chaque service parmi une pluralité de services.

4. Procédé de gestion selon la revendication 3, caractérisé en ce que ledit bloc d'information représentative des besoins de qualité de service comporte une séquence de dimensions de QdS, chaque dimension de QdS contenant une séquence de Domaines, et chaque Domaine contenant une séquence d'attributs de QdS.

5. Procédé selon la revendication 2, caractérisé en ce que pour rassembler les capacités de prestations des ressources et/ou services à partir des applications prestataires de ressources et/ou de services, on utilise une structure d'information dynamique représentative des offres de qualité de service à l'attention des utilisateurs, 5 comprenant un en-tête de contexte de fournisseur de service, un bloc d'information représentatif des spécifications de qualité de service offertes par chaque service disponible, et un un bloc d'information représentatif des spécifications de qualité de service pour chaque service parmi une pluralité de services.

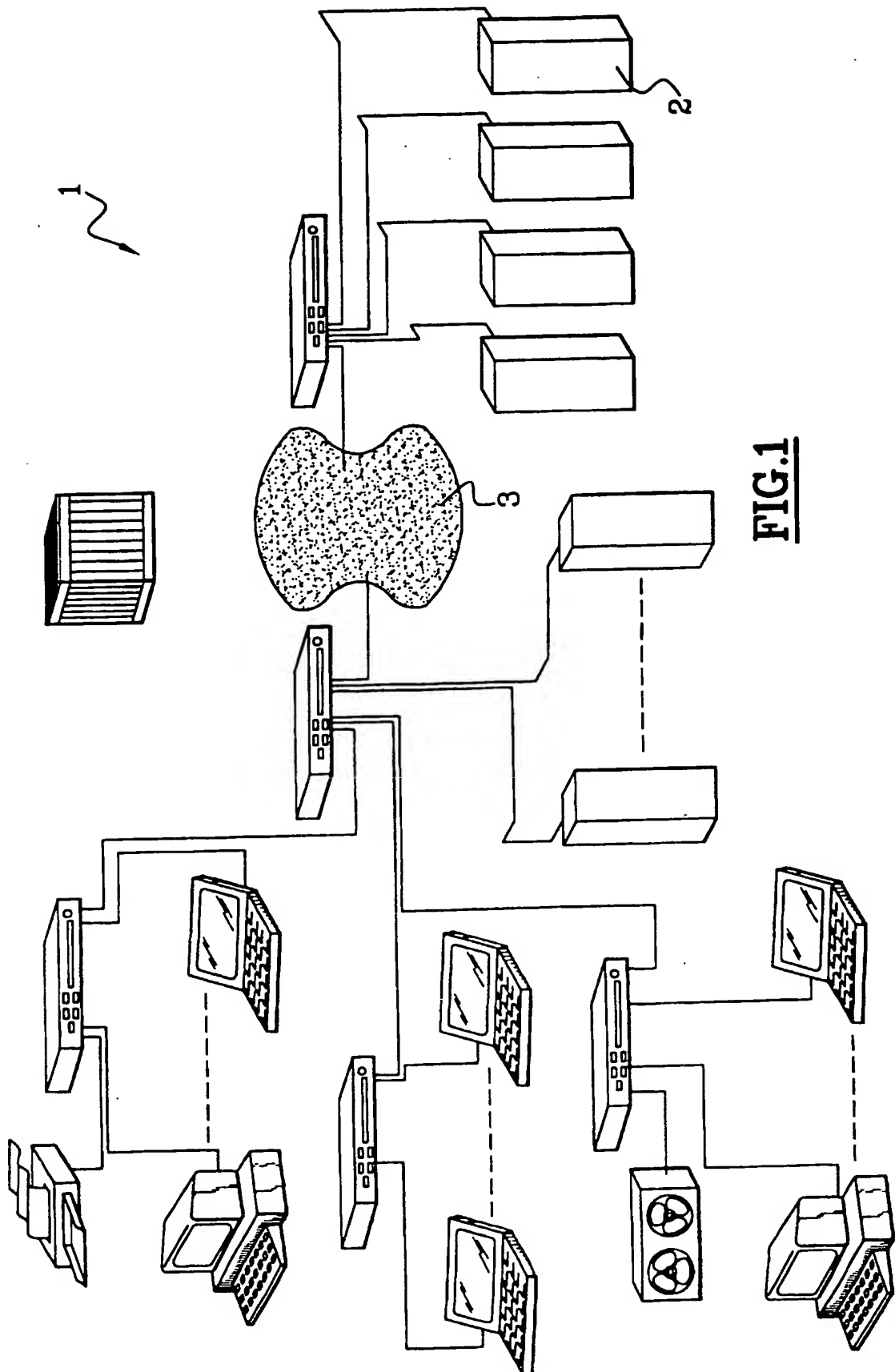
6. Procédé selon l'une quelconque des revendications précédentes, 10 caractérisé en ce que pour traiter les informations reçues et décider des opérations de gestion à effectuer, on élabore des tâches de gestion abstraites à l'aide d'agents autonomes qui prennent en compte les besoins de qualité de servie QoS des utilisateurs, et les offres de service dans l'environnement du réseau.

7. Procédé selon la revendication 6, caractérisé en ce que lesdits agents 15 sont des agents intelligents, comportant :

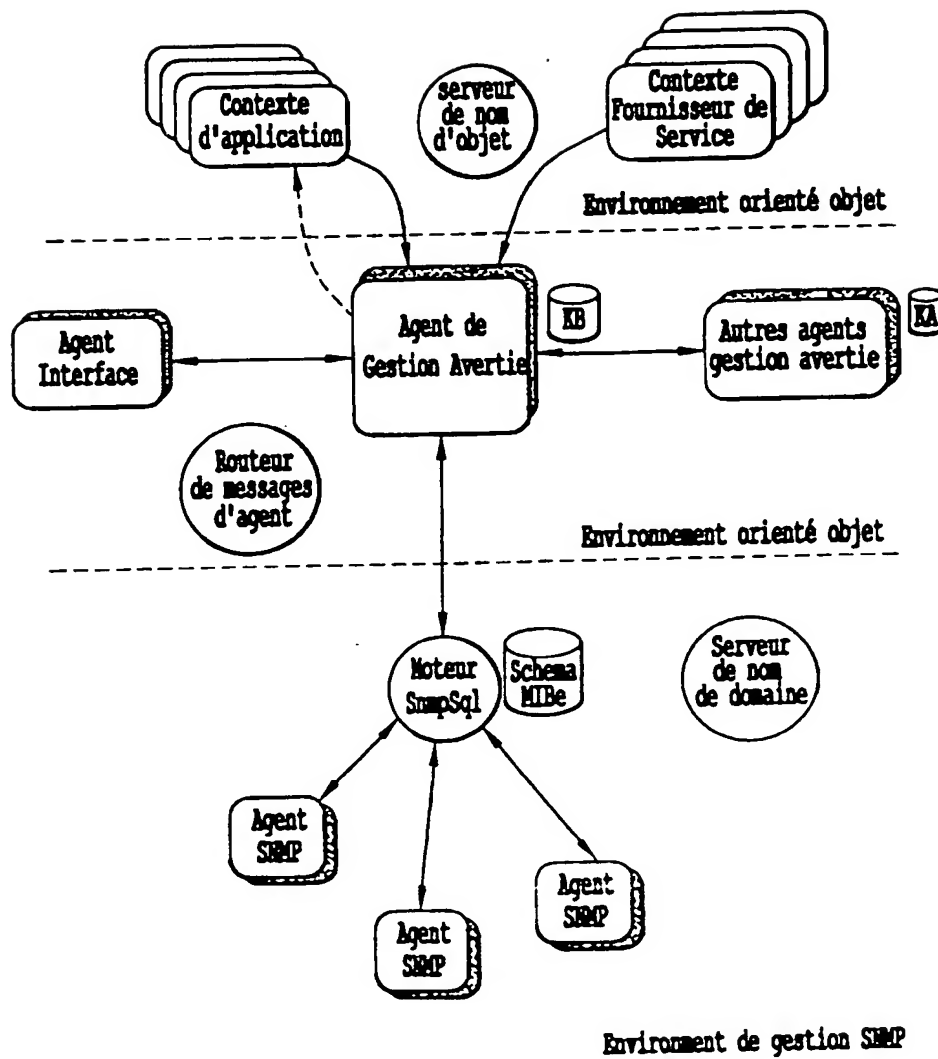
- une couche avertie permettant d'acquérir les besoins des utilisateurs du réseau et les offres de services;
- une couche délibérative chargée de créer des tâches de gestion;
- une couche opérationnelle exécutant les tâches de gestion créées par la 20 couche délibérative.

8. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que la saisie des besoins des utilisateurs se fait par l'intermédiaire d'une interface homme-machine comprenant un moyen d'entrée d'information et un moyen d'affichage.

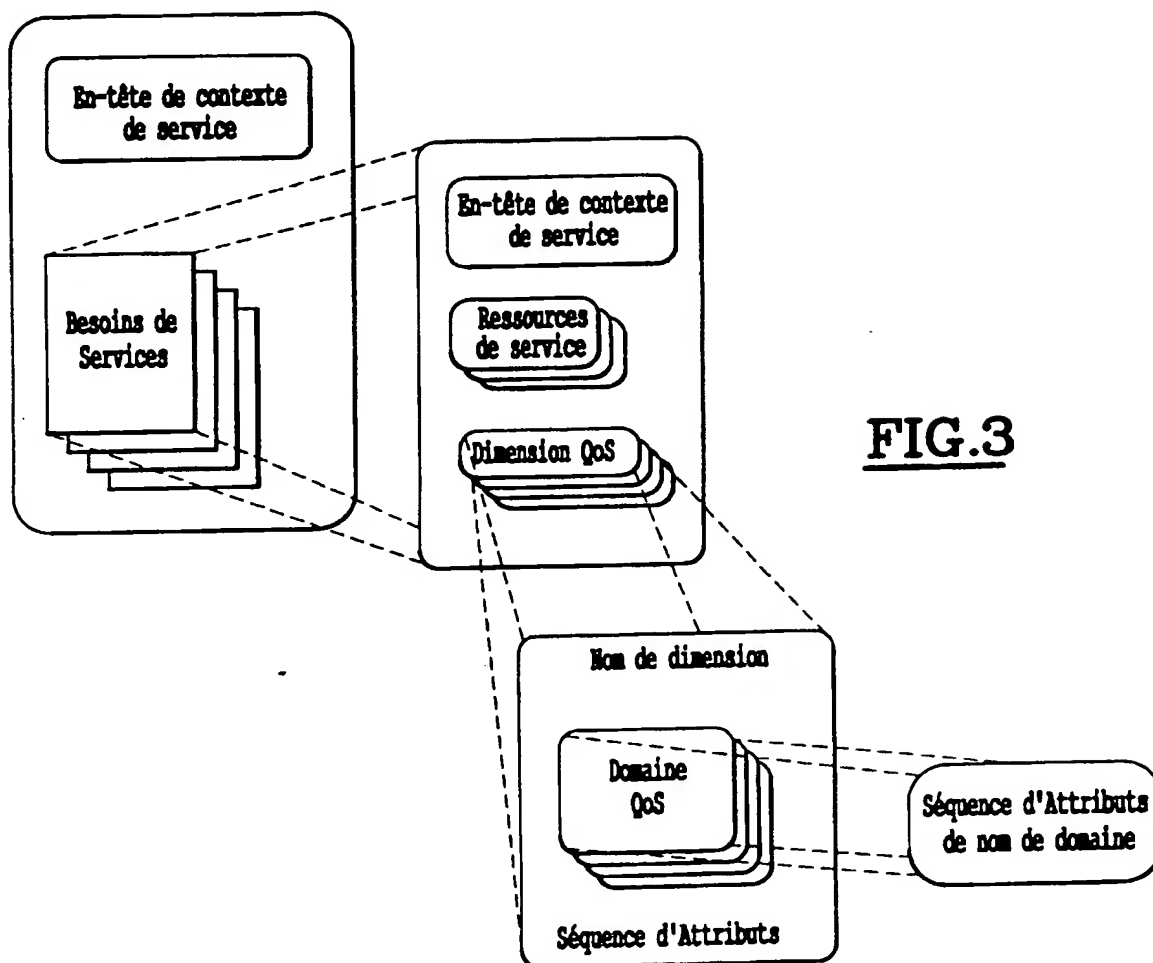
1/7

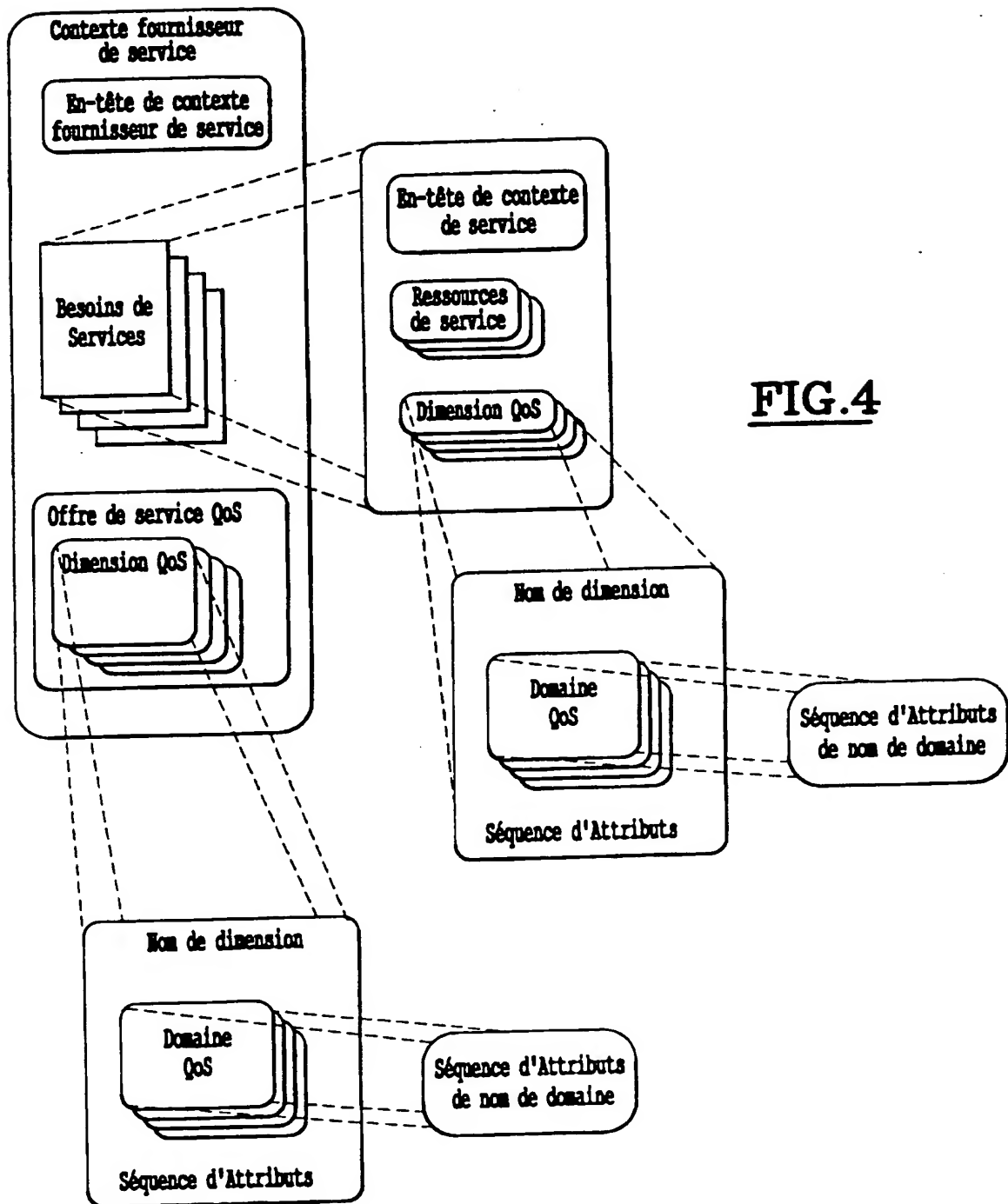
**FIG. 1**

2/7

**FIG.2**

Contexte d'application



**FIG.4**

5/7

FIG.5**Contexte d'application****0 En-tête de contexte d'application**

```

:X Session
:Session utilisateur
:oliveira
:501B
:Violette.eurocom.fr
:10

```

Contexte de service**1 En-tête de contexte de service Dimension de QoS**

```

:Dfs
:omegalab.eurocom.fr
:50
:données
:meilleurs efforts

```

1.1

```

:Dfs à temps
:100

```

1.1.1 Domaine QoS

```

:Dfs à temps
:100
:(temps de réponse, (0,10,long), (ms,1,1))
:(nombre de requêtes, (10,10,long),)
:(temps de réponse moyen, (0,5,long), (ms,1,1))

```

Contexte de service**2 En-tête de contexte de service Dimension de QoS**

```

:Dprinting
:esteron.eurocom.fr
:50
:données
:meilleurs efforts

```

2.1

```

:fiabilité Dfs
:100
:(period, (3600,3600,long), (s,1,1))

```

2.1.1 Domaine QoS

```

:fiabilité disponibilité Dfs
:100
:(temps moyen de réparation, (0,300,long), (s,1,1))
:disponibilité, (95,100,int),)

```

2.1.2 Domaine QoS

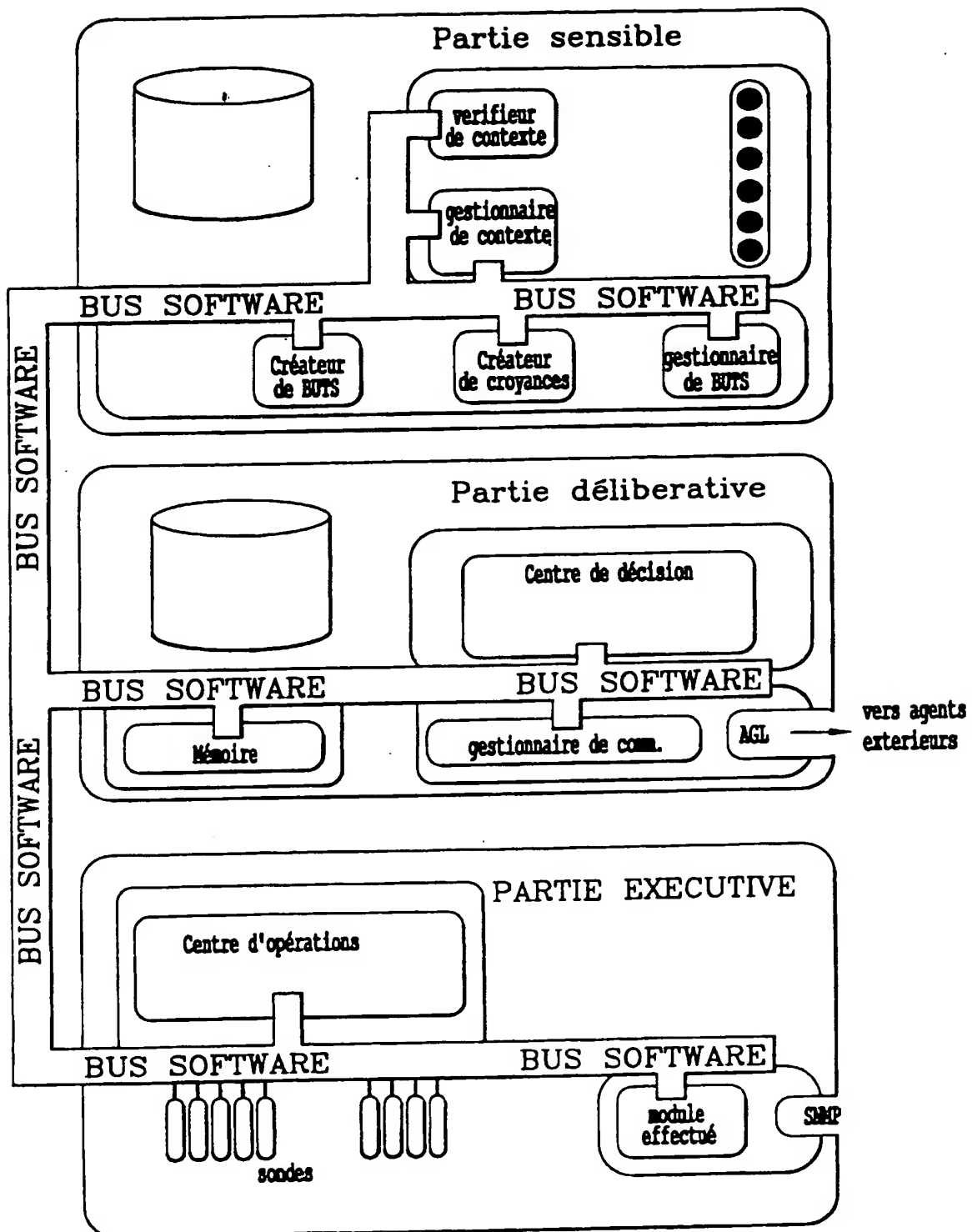
```

:fiabilité Dfs fiabilité Dom
:100
:(temps moyen entre pannes, (18000,,long), (s,1,1))
:(nombre de pannes, (0,1,long), scalar)

```

Séquence physique de contexte : 0 : 1 : 1.1 : 1.1.1 : 2 : 2.1 : 2.1.1 : 2.1.2

6/7

**FIG.6**

7/7

